Game-theoretic Management of Interacting Adaptive Systems

David Wolpert
NASA Ames Research Center
MS 269-1
Moffett Field, CA 94035
Email: david.h.wolpert@nasa.gov

Nilesh Kulkarni
Perot Systems INC, NASA Ames Research Center
MS 269-1
Moffett Field, CA 94035
Email: nilesh.v.kulkarni@nasa.gov

***Abstract*— A powerful technique for optimizing an evolving system "agent" is co-evolution, in which one evolves the agent's environment at the same time that one evolves the agent. Here we consider such co-evolution when there is more than one agent, and the agents interact with one another. By definition, the environment of such a set of agents defines a non-cooperative game they are playing. So in this setting co-evolution means using a "manager" to adaptively change the game the agents play, in such a way that their resultant behavior optimizes a utility function of the manager. We introduce a fixed-point technique for doing this, and illustrate it on computer experiments.**

I. I

Many distributed systems involve multiple goal-seeking agents. Often the interactions of those agents can be modeled as a noncooperative game where the agents are identified with the players of the game and their goals are identified with the associated utility functions of the players [7], [8], [12], [17], [21]. Examples involving purely artificial players include distributed adaptive control, distributed reinforcement learning (e.g., such systems involving multiple autonomous adaptive rovers on Mars or multiple adaptive telecommunications routers), and more generally multi-agent systems involving adaptive agents [2], [6], [15], [16]. In other examples some of the agents / players are human beings. Examples include air-traffic management [9], multi-disciplinary optimization [4], [5], and sense, much of mechanism design, including in particular design of auctions [8], [12], [13].

Sometimes the goals of the players do not conflict; intuitively, the system is "modularized". However often this cannot be guaranteed. As an example, it may be that in most conditions the system is modularized, but that some conditions cause conflicts among the needs of the players for system-wide resources (e.g., when an "emergency" occurs). Alternatively, it may be that the players take physical actions and that under some conditions the laws of physics couple those actions in way that makes their goals conflict. Moreover, whenever some agents are humans, in almost all conditions there will be some degree of conflict among their goals. Finally, note that even when there are no conflicts among the goals of the players, there may be synergies among the players that they can not readily find if left on their own.

In all of these scenarios there is a need to intervene in the behavior of the players. However often we do not have complete control over the behavior of the players. That is always true if there are communication restrictions that prevent us from having full and continual access to all of the players. It is also always true if some of the players are humans. Such limitations onour control are also typical when the players are software programs written by third party vendors.

On the other hand, even when we cannot completely control the players, often we can set / modify some aspects of the game among the players. As examples, we might have a manager external to the players who can modify their utility functions (e.g., by providing them with incentives), the communication sequence among them, what they communicate with one another, the command structure relating them, how their chosen moves are mapped into the physical world, how (if at all) their sensory inputs are distorted, or even how rational they are. The ultimate goal of the manager is to make such modifications that induce behavior of the players that is optimal as far as the manager is concerned.

As an example, say some of the players are evolvable software systems. Then the game details comprise the environment in which those systems evolve. So modifying the game details to optimize the resultant behavior of the players is a variant of using co-evolution for optimization [1], [3]. The difference with most work on co-evolution is that in optimal management of a game we are concerned with the environment of multiple interacting and evolving systems rather than the environment of a solitary evolving system.

In the next section we present a simple example that illustrates how the behavior of players in a game can be improved by changing the game, discuss previous related work, and overview our approach. After that we present our notation. We then use that notation to introduce a formal framework for analyzing optimal management. Next we use our framework to introduce an algorithm for optimal management. After that we present a computer-based test validating that algorithm.

II. B

To illustrate how changing the details of a game may result in the players behaving in a way that is better for an external manager, say we have two players, *Row* and *Col*, each of who has four possible moves (also called "pure strategies"). As usual, each player has a utility function that maps the joint

pure strategy of the two players into the reals. Say that those utility functions, (g^R, g^C), are given by the following bimatrix:

$$\begin{bmatrix} (0,6) & (4,7) & (-1,5) & (4,4) \\ (-1,6) & (5,5) & (2,3) & (7,4) \\ (-2,1) & (3,2) & (0,0) & (5,-1) \\ (1,1) & (6,0) & (1,-2) & (6,-1) \end{bmatrix} \quad (1)$$

To play the game each player $i \in \{\text{Row, Col}\}$ independently chooses a "mixed strategy", i.e., a probability distribution $P_i(x_i)$ over her set of (four) allowed moves. So the expected utility for player i is $\mathbb{E}_P(g^i) = \sum_{x_i, x_{-i}} P_i(x_i) P_{-i}(x_{-i}) u(x_i, x_{-i})$, where $P_{-i}(x_{-i})$ is the mixed strategy of i's opponent.

A pair of mixed strategies (P_{Row}, P_{Col}) is a "Nash Equilibrium" (NE) of the game if for all players i, $\mathbb{E}_P(g^i)$ cannot increase if P_i changes while P_{-i} stays the same. At a NE, neither player can benefit by changing her mixed strategy, given her opponents' mixed strategies. To illustrate, a NE of the game in Table 1 is the joint pure strategy where Row plays her bottom-most move, and Col plays her left-most move. Noncooperative game theory's starting premise is that "rational" players will have a NE joint mixed strategy.

Now say that we could induce both players to be "anti-rational", that is to try to *minimize* their expected utilities. (Formally, this is equivalent to multiplying their utility functions by -1.) Now the equilibrium of the game occurs where Row plays the top-most row and Col plays the right-most column. Note that both players have a higher utility at this equilibrium than at the NE ($4 > 1$). Moreover, neither player would benefit if she changed from being anti-rational to being rational and the equilibrium changing accordingly, regardless of which rationality her opponent adopted. Accordingly, consider the scenario where the manager's goal is to increase the utility functions of both players. Then if he can infer that joint anti-rationality does that, and can also induce the players both to be anti-rational, the manager would benefit.

More generally, the optimal management problem is how to find changes that the manager can make to the system that would make it behave in a way that the manager prefers. In the Probability Collectives [14], [19], [20] approach to optimal management, one has complete freedom to design the players in some subset T of all the players. The remaining players are treated as exogenous noise, with no attempt to exploit prior knowledge that they are goal-seeking. Distributed reinforcement learning and related approaches share these characteristics [2], [6], [15], [16]. In the Collective Intelligence approach [21], [22] one only has freedom to design the utility functions of the players in T, and again treats all other players as exogenous noise. Most of mechanism design [8], [12], [13] shares these characteristics (though in mechanism design, one talks of "conditional payments" to players rather than "modifications to their utility functions").

In contrast, here we focus on a more general situation. We expect that manager may only have partial ability to modify the players in T, and that those modifications may involve other characteristics besides their utility functions. We also allow the manager to change system parameters not directly part of any player. Finally, the players not in T are modeled as players, rather than treated as noise.

Our approach starts by specifying a parameterized set of models for the probability distribution of the entire system. To capture our prior knowledge that the players are goal-seeking, these models are based on game theory considerations. There are two types of parameters of our models: those that characterize the behavior of the players, and those that the manager sets. The former are estimated from observations of system behavior. The manager then uses those estimates and searches over the remaining parameters, to find which of the associated probability distributions are optimal for him. He then sets those parameters to their optimizing values.

III. G

Given any space Z, we write the set of functions from Z into Z as Z^Z. We use a minus sign before a set of subscripts of a vector to indicate all components of the vector other than the indicated one(s). We will use the integral symbol with the measure implicit. So for example for finite X, "$\int dx$" implicitly uses a point-mass measure and therefore means a sum. We write $|Z|$ to indicate the cardinality of Z.

We use upper cases to indicate a random variable, and lower cases to indicate a value of that variable. So for example "$P(A \mid b)$" means the full function mapping any value a of the random variable A to the conditional probability of a given the value b of the random variable B. We will be loose in distinguishing between probability distributions and probability density functions, using the term "probability distribution" and the symbol P to mean both concepts, with the context making the precise meaning clear if only one of the concepts is meant. We will also use the Dirac delta symbol even if its arguments are integers or even symbols. So for example the expression $\delta(a - b)$ where a and b are members of an arbitrary finite space equals 1 if $a = b$, 0 otherwise.

We write $\mathbb{N}$ to mean the infinite set of all integers not less than 1, and $\mathcal{N} \equiv 1, \ldots, N$. We will sometimes use curly brackets to indicate a set of indexed elements without explicitly specifying the range of the indices. For example, in an N-player game where each player i has an associated variable a_i, "$\{a_i\}$" is shorthand for $\{a_i : i \in \mathcal{N}\}$. Similarly a_{-j} is shorthand for $\{a_i : i \in \mathcal{N}, i \neq j\}$. $sgn(x)$ is defined to equal the sign of the real value x, with $sgn(0) \equiv 0$.

When defining a function, the symbol "$\triangleq$" means the definition holds for all values of the listed arguments. So for example, "$f(a, b) \triangleq \int dc\, r(a) s(b, c)$" means the definition holds for all values of a and b (as opposed to being an equation whose solution specifies a and b). Abusing notation, given a function $F : A \to B$, we will sometimes write $F(A)$ to indicate the range of F, but sometimes write $F(A')$ to indicate the full function F evaluated over domain A'.

The unit simplex of possible distributions over a space Z is written Δ_Z. Given two spaces A, B, we write $\Delta_{A,B}$ to mean the unit simplex over the Cartesian product $A \bigtimes B$. Similarly, $\Delta_{A|b}$ indicates the set of all distributions of A conditioned on b, and $\Delta_{A|B}$ indicates the set of all functions from B into Δ_A, i.e, the

set of all conditional distributions $P(A \mid b)$, $b \in B$. Finally, given a Cartesian product space $X = \bigtimes_i X_i$, we write Δ_χ to indicate the Cartesian product $\bigtimes_i \Delta_{X_i}$. So Δ_χ is the set of all product distributions over X. Similarly, $\Delta_{\chi|a}$ is $\bigtimes_i \Delta_{X_i} \mid a$, the set of all product distributions over X conditioned on a.

IV. T

A. Exact State Information

Say we have a repeated game, with the set of possible system states (states of Nature) given by Θ, and the set of possible joint moves by the players given by X [8], [12]. For simplicity, say this is a stationary game of complete state information [12]. So we can dispense with a special space of possible signals to the players — at every iteration every player knows the current state of Nature exactly. In fact, we restrict attention to behavior strategies by the players that only depend on that preceding state of Nature; every player's behavioral strategy is a conditional distribution over the possible moves of that player conditioned on the preceding state of Nature. Also for simplicity, we take Θ and X finite.

We write the (fixed) conditional probability of the next state given the current state and the next joint move as

$$\pi(\theta_{t+1}, \theta_t, x_{t+1}) \triangleq P(\theta_{t+1} \mid \theta_t, x_{t+1}) \tag{2}$$

where $t \in \mathbb{N}, \theta_{t+1}, \theta_t \in \Theta$ and $x_{t+1} \in X$. The joint behavioral strategy of the players at time t is the distribution

$$\begin{aligned} P(x_{t+1} \mid \theta_t) &= \prod_i P(x^i_{t+1} \mid \theta_t) \\ &\triangleq \prod_i \sigma^i_t(x^i_{t+1}, \theta_t). \end{aligned} \tag{3}$$

(Note that it is a matter of conventional whether we have the index on x be t or $t+1$; we choose it to be $t+1$ so that the behavioral strategy of the players is formulated as the move the players will make given the system state they just observed.)

Intuitively, each component x^i of x is set independently of the other components by player i, in a completely free manner. It is in how those moves affect the next state of Nature that "constraints" can arise that physically couple the devices controlled by the players. In general, we place no restrictions on the form of each $P(X^i_{t+1} \mid \Theta_t)$; they are to be set solely by associated game-theoretic considerations.

Given this, the updating rule for the distribution over Θ is

$$P(\theta_{t+1}) = \int d\theta_t dx_{t+1}\, \pi(\theta_{t+1}, \theta_t, x_{t+1}) \prod_i \sigma^i_t(x^i_{t+1}, \theta_t) P(\theta_t) \tag{4}$$

This is the transition equation for a Markov process taking $P(\Theta_t)$ to $P(\Theta_{t+1})$, where the transition matrix $\int dx_{t+1}\ \pi(\Theta_{t+1}, \Theta_t, x_{t+1}) \prod_i \sigma^i_t(x^i_{t+1}, \Theta_t)$ is parameterized by the behavioral strategies of the players. Write this matrix as $A(\Theta_{t+1}, \Theta_t; \sigma_t)$, or $A(\sigma_t)$ for short, where σ_t is the vector of all players' behavioral strategies at time t. Similarly write the update equation Eq. 4 as $P(\Theta_{t+1}) = A(\sigma_t)P(\Theta_t)$.

For simplicity, we restrict attention to scenarios where at any moment t, the goal of each player i is to maximize an expected associated utility $g^i : \Theta \to \mathbb{R}$ evaluated at time $t+1$. So our players are single-step receding horizon controllers. (An example of an alternative is where each player wants to maximize the expectation of a discounted sum of future rewards.) Expanding, we write that expected utility as

$$\begin{aligned} \mathbb{E}[g^i(\Theta_{t+1})] &= \int d\theta_{t+1}\, P(\theta_{t+1}) g^i(\theta_{t+1}) \\ &= \int d\theta_t \left[\int d\theta_{t+1}\, A(\theta_{t+1}, \theta_t; \sigma_t) g^i(\theta_{t+1}) \right] P(\theta_t). \end{aligned} \tag{5}$$

Now plug in the definition of A to get

$$\begin{aligned} \mathbb{E}[g^i(\Theta_{t+1})] &= \int d\theta_t\, P(\theta_t) \left[\int dx_{t+1} \prod_j \sigma^j_t(x^j_{t+1}, \theta_t) \times \right. \\ &\qquad \left. \int d\theta_{t+1}\, \pi(\theta_{t+1}, \theta_t, x_{t+1}) g^i(\theta_{t+1}) \right] \\ &\triangleq \int d\theta_t\, P(\theta_t) \int dx_{t+1} \prod_j \sigma^j_t(x^j_{t+1}, \theta_t) \gamma^i(x_{t+1}; \theta_t) \end{aligned} \tag{6}$$

where for all i, $\gamma^i(x_{t+1}; \theta_t) \triangleq \mathbb{E}(g^i_{t+1} \mid x_{t+1}, \theta_t)$.

This shows that each value of θ_t specifies a separate conventional strategic form game for the players, with their θ_t-specific mixed strategies over X_{t+1} given by $\{\sigma^i_t = P(X^i_{t+1} \mid \theta_t) : i \in \mathcal{N}\}$ and their θ_t-specific "effective" utility functions over X_{t+1} given by $\{\gamma^i(X^i_{t+1}; \theta_t) : i \in \mathcal{N})\}$. The actual time t is irrelevant to the specification of the game; only the value of the parameter θ_t matters, by setting the dependence of the effective utility functions on X_{t+1}. We write this game jointly specified by θ_t, π and the set $g \triangleq g^i\}$ as the **game function** $B^{\pi,g}(\theta_t)$, or sometimes just $B(\theta_t)$ for short. (So for every θ_t, $B(\theta_t)$ is a set of N utility functions, $\{\gamma^i(X; \theta_t)\}$.)

Accordingly, consider the following iterated process. First, at time t we sample $P(\Theta_t)$ to get a θ_t. Next, the players set $\sigma_t(X_{t+1} \mid \theta_t)$ to a NE of the game $B(\theta_t)$. Each player i then samples her mixed strategy $\sigma^i_{t+1}(X^i_{t+1} \mid \theta_t)$ to get a move x^i_{t+1}. This specifies a joint move x_{t+1}. After this $\pi(\Theta_{t+1}, \theta_t, x_{t+1})$ is sampled to get a next Θ value, θ_{t+1}, and the process repeats.

Now in general, there may be more than one NE for some game $B(\theta)$. More broadly, if the players are allowed to have bounded rationality, the set of possible joint mixed strategies adopted by the players for any game $B(\theta)$ may have non-zero measure. However say we have a **universal refinement** which the players jointly use to always pick the same unique joint distribution for any game, i.e., a mapping $R : B(\Theta) \to \Delta_\chi$. (As an example, it might that for any θ, $R(B(\theta))$ is a NE of game $B(\theta)$.) Then $\sigma_t(X_{t+1}, \theta_t) = R(B(\theta_t))$.[1] This means that for fixed π and g, σ_t is a t-independent function from Θ to Δ_χ (changes to t that don't θ_t don't change $B(\theta_t)$ and therefore don't change $\sigma_t(X_{t+1}, \theta_t)$.) Accordingly we can drop the subscript t from σ_t.

R induces a (t-independent) conditional distribution

$$P^{b,R}(\theta_{t+1} \mid \theta_t) \triangleq \int dx_{t+1}\, \pi(\theta_{t+1}, \theta_t, x_{t+1})\, [R(b)](x_{t+1}) \tag{7}$$

[1] Such "point prediction" specifying a single possible σ for a given π and g, is the goal of conventional game theory. More sophisticated modeling provides a distribution over σ's. See [18].

where $b \in B(\Theta)$. (Note that $P^{b,R}$ is well-defined even if $b \neq B(\theta_t)$.) If we sample this conditional distribution for a current pair $(\theta_t, b = B(\theta_t))$ we (stochastically) generate a new Θ value, θ_{t+1}. Evaluating $B(\theta_{t+1})$ then produces a new game b, and we can then apply Eq. 7 using that new game, so that the process repeats. In this way, for any fixed R and π, the function $B(\Theta)$ generates a distribution over possible sequences of θ's.

An interesting special case is when B is invertible, i.e., where knowing the N function $\{\gamma^i(X_{t+1}; \theta_t)\}$ uniquely fixes θ_t. In such cases we can dispense with Θ; the stochastic dynamics of the system across Θ reduces to a stochastic dynamics across an associated space of games, $B(\Theta)$.[2]

B. Partial state information

Now say that each player i not see θ_t at iteration t, but only a "signal" w_t^i that is stochastically related to θ_t. So the updating rule for the distribution over Θ is now

$$P(\theta_{t+1}) = \int d\theta_t dx_{t+1} dw_t \; \pi(\theta_{t+1}, \theta_t, x_{t+1}) \times \left[\prod_i [h_t^i(x_{t+1}^i, w_t^i)\Omega^i(w_t^i, \theta_t)\right] P(\theta_t) \quad (8)$$

where $h_t^i(x_{t+1}^i, w_t^i) \triangleq P(x_{t+1}^i \mid w_t^i)$ replaces σ^i as the strategy adopted by player i at time t, and where $P(w \mid \theta) \triangleq \prod_{i \in \mathcal{N}} \Omega^i(w^i, \theta)$ is the (fixed) conditional distribution specifying how the vector of all the signals of all players, w, is stochastically generated from a current state of nature θ.[3]

Now by Bayes' rule, we can always expand

$$\begin{aligned} P(w_t)P(\theta_t \mid w_t) &= P(\theta_t)P(w_t \mid \theta_t) \\ &= P(\theta_t) \prod_{i \in \mathcal{N}} \Omega^i(w_t^i, \theta_t) \end{aligned} \quad (9)$$

Plugging this into Eq. 8,

$$\begin{aligned} \mathbb{E}[g^i(\Theta_{t+1})] &= \int dw_t \; P(w_t) \int dx_{t+1} \prod_j h_t^j(x_{t+1}^j, w_t^j) \times \left[\int d\theta_t d\theta_{t+1} P(\theta_t \mid w_t)\pi(\theta_{t+1}, \theta_t, x_{t+1}) g^i(\theta_{t+1})\right] \\ &\triangleq \int dw_t P(w_t) \int dx_{t+1} \prod_j h_t^j(x_{t+1}^j, w_t^j) r_t^i(x_{t+1}; w_t) \end{aligned} \quad (10)$$

where the value of the vector w at time t is w_t and where each $r_t^i(X_{t+1}; W_t)$ is an "effective" utility function for player i. The full-rationality equilibrium at time t is any set of strategies $\{h_t^j\}$ such that simultaneously each h_t^i maximizes the expected (effective) utility, $\mathbb{E}(r_t^i(\Theta_t; W_t))$, as evaluated using h_t^{-i}.

subject to the strategies of the other players.

Recall that when every player has exact knowledge of the current state, the players are involved in a θ_t-indexed strategic form game, by Eq. 6. So comparing Eq. 6 with Eq. 10, it might seem that in the partial state information case the players are involved in "a w_t-indexed strategic form game". This is not strictly correct though. The problem with the comparison is that whereas each h_t^i only depends on w^i, the effective utility r_t^i depends on the entire vector w; there is no such distinction in the arguments of the corresponding functions in Eq. 6. Here player i can only use w^i to choose her move, whereas her utility function depends on the full w.

More carefully, what Eq. 10 really shows is that at each time t the players are involved in a correlated equilibrium game.[4] Moreover, since each conditional distribution Ω^i is fixed in time, the distribution $P(W_t)$, along with each effective utility function r_t^i, all vary with $P(\Theta_t)$.[5] So the correlated game the players are engaged in varies with t in general, and therefore so does the full rationality equilibrium h_t.

To understand this intuitively, note that $P(\Theta_t)$ changes as the system evolves, just as θ_t changes in the exact information case. (See Eq. 8.) Moreover, in this partial information setting $P(\Theta_t)$ is a "prior probability" that each player i uses to infer what θ_t is likely to be, having observed signal w_t^i. Accordingly, changes to $P(\Theta_t)$ changes the optimal behavior strategies h_t^j.

Just as θ_t defines the game of Eq. 6, so $P(\Theta_t)$ defines the game of Eq. 10. In both cases, the game the players are engaged in changes in time. In the partial information case, the system evolves from one t to the next by going through a sequence of correlated equilibria. For every t, the equilibrium of the associated game specifies a new game whose correlated equilibrium gives the time t+1 joint move, x_{t+1}. The associated transition rule is non-Markovian, i.e., since the optimal h_t depends on $P(\Theta_t)$ which changes in time, the transition rule in Eq. 8 is not a Markovian process over $X \times \Theta$. In particular, the dynamics over Θ is not governed by the matrix A.

For this partial information case, the game function has $P(\Theta_t)$ as its argument rather than θ_t (the argument of the game function in the exact information case), and is parameterized by Ω in addition to π and g. Similarly, the domain of any universal refinement for the partial information case is

[2] One example of an invertible B is presented in the experiments below in which the two players have different utility functions and there is no manager, so the thruster angles are fixed. As a more extreme example, let Θ be a subset of $\mathbb{R}^N$, and define each g^i as the associated projection operator, $g^i(\theta) \triangleq \theta^i$. Presume further that for all $x_{t+1} \in X, \theta_t \in \Theta, \theta_{t+1} \in \Theta$, $\pi(\theta_{t+1}, \theta_t, x_{t+1}) = \delta(\phi(\theta_t, x_{t+1}) - \theta_{t+1})$ for some vectgor-valued function ϕ. So for all i, θ_t and x_{t+1}, $\gamma^i(x_{t+1}; \theta_t) = \phi^i(\theta_t, x_{t+1})$. Now ϕ takes $X \times \Theta \to \Theta$, i.e., X acts on Θ via ϕ. Say that each $x^i \in X^i$ is a different permutation of Θ^i. Then not only is $B(\Theta_t)$ invertible, but in fact we only need to know the N values $\{\gamma^i(x_{t+1}; \theta_t)\}$ for one x_{t+1} to know θ_t.

[3] There are many variations of this scenario. For example, we could have each player i base her probability of move x^i at some time t on a history of values $w_{t'}^i$ for $t' < t$ in addition to w_t^i. Many such variations can be mapped into one another by redefining what the variables mean.

[4] One can see this by considering a two-stage extensive form game based on our original strategic form game. In that two-stage game there is a single first-moving Nature player who sets w_t by playing the (potentially time-varying) distribution $P(W_t)$. The original players $\mathcal{N}$ simultaneously move in the second stage, and each such player $i \in \mathcal{N}$ has an information set consisting of the value w_t^i. The utility function of the i'th player in the two-stage game is the associated effective utility function of player i in the original game, r_t^i. Now if $h(X_{t+1}, W_t^j)$ is a full rationality equilibrium of the original game, then there is no player i and function ψ such that $\int dw_t dx_{t+1} \; P(w_t)h_t(x_{t+1}, w_t^j)r_t^i(\psi(x_{t+1}^i), x_{t+1}^{-i}; w_t) > \mathbb{E}[g^i(\Theta_{t+1})]$. So that h_t is a correlated equilibrium of the two-stage game.

[5] Integrate both sides of Eq. 9 over θ_t to see that $P(W_t)$ depends on $P(\Theta_t)$. Then use Eq. 9 again to see that $P(\Theta_t \mid W_t)$ depends on $P(\Theta_t)$, and therefore so does each r_t^i.

an expanded version of the domain for exact information univeral refinements, now being the set of all correlated equilibrium games. Given such a partial information game function $\bar{B}^{\pi,g,\Omega}$ and refinement $\bar{R}$, we can write $h_t(x_{t+1}, w_t) = \bar{R}[\bar{B}^{\pi,g,\Omega}(P(\Theta_t))](x_{t+1}, w_t)$. Plugging this into Eq. 8 then gives the dynamics over Θ:

$$P^{\pi,g,\bar{R},\Omega}(\theta_{t+1}) = \int d\theta_t dx_{t+1} dw_t \; \pi(\theta_{t+1}, \theta_t, x_{t+1}) \Omega(w_t, \theta_t) \times \left[\bar{R}[\bar{B}^{\pi,g,\Omega}(P(\Theta_t))](x_{t+1}, w_t)\right] P(\theta_t) \quad (11)$$

where $\Omega(w_t, \theta_t) \triangleq \prod_i \Omega^i(w_t^i, \theta_t)$.

V. O M

A. General considerations of optimal management

From now on, for simplicity we restrict attention to the case where the players have exact state information, as in Sec. IV-A. This means that we do not consider the effects of sensor noise on player behavior.

Say we have a **manager** external to the players who at each t has a preference order over sequences of future θ's. So the manager prefers π's, R's, and g's that are more likely to produce desirable sequences of θ's, as determined by Eq. 7:

$$P^{\pi,g,R}(\theta_{t+1} \mid \theta_t) \triangleq \int dx_{t+1} \; \pi(\theta_{t+1}, \theta_t, x_{t+1}) \; [R(B^{\pi,g}(\theta_t))](x_{t+1}) \quad (12)$$

(For the partial information case, dynamics over Θ is instead given by the non-Markovian Eq. 11.)

Suppose that the manager can change some aspects of π and/or g and/or R. Then at each t the manager has an optimization problem, of how to choose among its set of triples $y \equiv (\pi, g, R)$ to optimize the likely resultant sequences of θ's. Note that since we are in the exact information case, we do not allow the manager to distort the sensor inputs to the players. (That would mean distorting Ω.) Similarly, it means we do not consider the possibility of the manager modifying the inter-player communication structure and/or command structure (i.e., we do not allow the manager to change the extensive game the players are engaged in).

As an example of this, the manager's preference order might be be a discounted sum of future rewards. This need not be the case however. To illustrate this, note from Eq. 12 that for any fixed y the transition matrix $A^y \triangleq P(\Theta_{t+1} \mid \Theta_t)$ is fixed for all time. So for any fixed y, we have a Markov process across $P(\Theta) \in \Delta_\Theta$ with transition matrix A^y and can analyze its convergence properties. In particular, the manager might prefer a sequence in Δ_Θ that eventually converges to a fixed point. (Note this is a fixed point in Δ_Θ, not in Θ.) Furthermore, if by setting y he can vary among a set of such fixed points $\{P(\Theta_\infty) \in \Delta_\Theta\}$, then his preference ordering might lead him to prefer ones that are centered about certain locations in Θ. He might also prefer a $P(\Theta_\infty)$ that is stable, in that it is highly peaked, so that in the infinite time limit ($P(\Theta_t)$ settles to a distribution under which) θ_t has little variability. In addition to these aspects of the fixed point $P(\Theta_\infty)$, the manager might prefer a fixed point that is stable under the Markovian dynamics. (This stability in the Markovian dynamics is different from the stability of a peaked fixed point; the first concerns Δ_Θ, and the second concerns Θ.) He might also prefer that the dynamics converges to the fixed point quickly from some initial distribution $P(\Theta_0)$.

Given any such preferences, the simplest version of the manager's optimization problem is to find the $y \in Y$ such that the associated Markov transition matrix A^y has a fixed point, and to optimize the location of that fixed point, its stability, and how quickly it is achieved. There are more sophisticated policies the manager might adopt however. For example, it may be that by judiciously "mode-switching" among the possible y as the system evolves, the manager can induce the dynamics to go from the initial $P(\Theta_0)$ to a desired fixed point $P(\Theta_\infty)$ more expeditiously than it would if any single $y \in Y$ were used for the entire sequence.

Note that despite having a preference ordering and a move to choose, the manager is not a player in the game. In particular, there is no dynamics of y. Rather the manager sets y from outside of the system.

B. Algorithm overview

For simplicity, from now on we restrict attention to the case where the preference order of the manager does not depend on the full future sequence through the space of games, involving fixed points, discounted sums of rewards, or something similar. Rather, like the players, at every t the manager is only interested in optimizing (the expectation of) an associated utility function of θ_{t+1}. This restriction means we do not need to explicitly consider θ-indexed effective utility functions, game functions, or the like. We write the manager's utilty function as $G : \Theta \to \mathbb{R}$.

In practice, the manager may not know all relevant attributes of the players and/or the rest of the system. In this case the manager must estimate those attributes at run-time. Since the underlying process is Markovian, this means taht the manager's problem is one of controlling a partially observable Markov decision process. However since here at every t the manager is only concerned with θ_{t+1} (rather than the whole future sequence of θ's), we adopt a simpler approach.

To begin, parameterize the triple $\{\pi, g, R\}$ that fixes Eq. 12 by (ζ, y). As before, y is the set of all parameters affecting the behavior of the players and the system that the manager can set. ζ is a set of other parameters outside of the manager's control that affect the behavior of the players and system, but can be estimated from observational data. These may include in particular parameters that characterize the endogenous behavior of the players. For example, ζ might specify the rationality of some player i (suitably quantified), or if the manager cannot modify g^i, ζ could specify g^i. Other components of ζ might affect multiple players at once, by modifying R, π and/or the parametric dependence of g on y. To formalize this we sometimes write $g^{\zeta,y}$, $R^{\zeta,y}$, and/or $\pi^{\zeta,y}$.

Any pair (ζ, y) specifies the dynamics over Θ, via Eq. 12. So presuming the manager's estimate of ζ is correct, since ζ is independent of y, the manager can determine

how varying y translates into variations in $\hat{\sigma}^{\zeta,y}(x_{t+1}, \theta_t) \triangleq R^{\zeta,y}[B^{\zeta,y}(\theta_t)](x_{t+1}, \theta_t)$. So the task for the manager is to find the y that maximizes

$$\mathbb{E}^y(G(\theta_{t+1}) \mid \theta_t) =$$
$$\int d\theta_{t+1} dx_{t+1}\, G(\theta_{t+1}) \pi^y(\theta_{t+1}, \theta_t, x_{t+1}) \prod_{i \in \mathscr{N}} \hat{\sigma}^{\zeta,y,i}(x^i_{t+1}, \theta_t) \quad (13)$$

Often the manager will not be able to find this optimal y. This may be due to ignorance of some of the distributions, computational limitations, inability to estimate some relevant components of θ_t, etc. More generally, it may be that the players are actually in a partial state information scenario, but the manager cannot solve for the y that optimizes $\mathbb{E}^{\zeta,y}(G_{t+1} \mid w_t)$ (e.g., due to ignorance of Ω, or of w_t).

In such situations we approximate how the joint behavorial strategy and system dynamics depends on y and ζ with an **equilibrium model**. This is a pair of counterfactual game and (perhaps bounded rational) refinement functions, $\bar{B}^{\zeta,y}$ and $\bar{R}^{\zeta,y}$. Our presumption is that if we change the integrand in Eq. 13 to involve those functions, the resultant dependence of $\mathbb{E}^{\zeta,y}(G_{t+1} \mid \theta_t)$ on (ζ, y) accurately approximates the true dependence.

In practice, the manager might also estimate $\hat{\sigma}^{\zeta,y'}(X_{t+1}, \theta_t)$ for the current θ_t from observations. Doing that allows the manager to avoid evaluating $\bar{R}^{\zeta,y}[\bar{B}^{\zeta,y}(\theta_t)]$, which typically would require solving a coupled set of equations. Intuitively, Nature solves the equations on behalf of the manager.[6] In this situation, the manager only has to solve for how the solution $\bar{R}^{\zeta,y}[\bar{B}^{\zeta,y}(\theta_t)$ would change if y were to change (for the given ζ. As illustrated below, this may reduce to the manager's estimating how the integrand in Eq. 13 varies with y, for the given ζ. From now on we suppress the ζ superscript.

C. Quantal Response Equilibria

We are interested in modeling players that are "bounded rational", i.e., who want to maximize their expected utilities but are not able to do so. A popular model for this situation is the Quantal Response Equilibrium (QRE) [10], [11]. Under the QRE, the mixed strategy of player i is a Boltzmann distribution over her move-conditioned expected utilities:

$$\sigma^i(x^i_{t+1}, \theta_t) \triangleq \frac{e^{\beta^i \mathbb{E}^y(g^{y,i}_{t+1} | x^i_{t+1}, \theta_t)}}{N^i(\theta_t)} \quad (14)$$

where N^i is the associated normalization constant,

$$N^i(\theta_t) \triangleq \int dx^i_{t+1}\, e^{\beta^i \mathbb{E}^y(g^{y,i}_{t+1} | x^i_{t+1}, \theta_t)}. \quad (15)$$

If at each t the objectives of the players involve future trajectories through Θ rather than just (as in this paper) θ_{t+1}, then the exponents in the QRE equations should be changed accordingly. Those exponents should also be changed if we are in a partial state information setting rather than an exact state information setting (by replacing each $\mathbb{E}^y(g^{y,i}_{t+1} \mid x^i_{t+1}, \theta_t)$, given by Eq. 6, with $\mathbb{E}^y(g^{y,i}_{t+1} \mid x^i_{t+1}, w^i_t)$, given by Eq. 10).

[6] Sometimes the manager can even solicit $\hat{\sigma}^{\zeta,y'}(X_{t+1}, \theta_t)$ from the players.

Note that the QRE mixed strategy for player i depends on the mixed strategies of the other players, through $\mathbb{E}^y(g^{y,i}_{t+1} \mid x^i_{t+1}, \theta_t)$. So the QRE is a set of coupled simultaneous equations. Brouwer's fixed point theorem guarantees that there is always at least one σ that solves this set of equations. Moreover, in the limit of $\beta^i \to \infty$, the QRE σ^i places zero probability mass on any move that doesn't maximize i's expected utility. Accordingly, as $\beta^i \to \infty$ for every player i, the QRE approaches a NE [10].

For each i, the associated QRE equations can be expanded as the $|X^i| + 1$ equations

$$\sigma^i(x^i_{t+1}, \theta_t) - \frac{e^{\beta^i \mathbb{E}^y(g^{y,i}_{t+1} | x^i_{t+1}, \theta_t)}}{N^i(\theta_t)} = 0 \quad \forall x^i_{t+1} \quad (16)$$
$$N^i(\theta_t) - \int dx^i_{t+1}\, e^{\beta^i \mathbb{E}^y(g^{y,i}_{t+1} | x^i_{t+1}, \theta_t)} = 0. \quad (17)$$

where for all i,

$$\mathbb{E}^y(g^{y,i}_{t+1} \mid x^i_{t+1}, \theta_t) =$$
$$\int d\theta_{t+1} dx^{-i}_{t+1}\, g^{y,i}(\theta_{t+1}) \pi^y(\theta_{t+1}, \theta_t, x^i_{t+1}, x^{-i}_{t+1}) \prod_{j \neq i} \sigma^j(x^j_{t+1}, \theta_t) \quad (18)$$

By plugging Eq. 18 into Eq.'s 16, 17 and running over all players i, we specify the QRE as a set of $M \equiv N + \sum_{i \in \mathscr{N}} |X^i|$ coupled simultaneous equations. For fixed θ_t, there are a total of M unknowns in those equations: the N normalization factors $\{N^i\}$ together with the $\sum_{i \in \mathscr{N}} |X^i|$ mixed strategy components of the players, $\{\sigma^i(x^i_{t+1}, \theta_t)\}$. We can condense those M equations in M unknowns into the following equation:

$$\mathbf{f}(\sigma, \mathbf{N}, y) = \mathbf{0} \quad (19)$$

where σ is the vector of $\sum_{i \in \mathscr{N}} |X^i|$ probabilities $\{\sigma^i(x^i_{t+1}, \theta_t) : i \in \mathscr{N}, j \in |X^i|\}$, $\mathbf{N}$ is the vector of N normalization factors, $\mathbf{0}$ is the M-dimensional vector of all 0's, and f is an M-dimensional vector-valued function. For any y, the solution to Eq. 19 for σ and $\mathbf{N}$ gives us $\hat{\sigma}^{\zeta,y}$ and the associated values $\{N^i\}$, respectively.

For simplicity, we model the player interaction as being a QRE for some suitable set of β_i's. (Exploring more sophisticated models is the subject of future work.) The set of the β^i's of all the players will comprise ζ in our experiments. In addition, every g^y will be independent of y; only π^y depends on y. The resultant dependence of the player mixed strategies on y is captured in Eq. 19.

D. Moving the QRE fixed point

The manager's expected utility is given by Eq. 13 where each $\hat{\sigma}^{\zeta,y,i}$ is given by Eq. 19. For a given ζ, the task of the manager is to move y so that the resultant σ solving Eq. 19 optimizes $\mathbb{E}^y(G(\theta_{t+1}) \mid \theta_t)$ as given by Eq. 13. In more detail, as the manager changes y, he changes the values in Eq. 18, which then changes the probabilities in Eq. 14. That in turn changes expected G, according to Eq. 13. The manager wants to search over y's to maximize this ensuing value of expected G. The manager can do this using a gradient descent over expected G based on the following equation:

$$\frac{\partial}{\partial y}\mathbb{E}^y(G(\theta_{t+1}) \mid \theta_t) =$$
$$\int d\theta_{t+1} dx_{t+1}\; G(\theta_{t+1}) \frac{\partial}{\partial y} [\pi^y(\theta_{t+1}, \theta_t, x_{t+1})] \prod_{i \in \mathcal{N}} \hat{\sigma}^{\zeta,y,i}(x^i_{t+1}, \theta_t)$$
$$+$$
$$\int d\theta_{t+1} dx_{t+1}\; G(\theta_{t+1}) \pi^y(\theta_{t+1}, \theta_t, x_{t+1}) \frac{\partial}{\partial y} \left[\prod_{i \in \mathcal{N}} \hat{\sigma}^{\zeta,y,i}(x^i_{t+1}, \theta_t) \right] \tag{20}$$

The first integral is what the manager's estimate of the gradient of expected G would be if the manager were a "conventional controller", who presumes that $\hat{\sigma}^{\zeta,y,i}(x^i_{t+1}, \theta_t)$ is some stationary distribution. The second integral is the correction term introduced if the manager accounts for the fact that the algorithms setting each q^i are actually adaptive players who (under the QRE model of their mutual adaptation) obey Eq. 19.

To compute the integrand terms in Eq. 20 involving partial derivatives of the $\hat{\sigma}^{\zeta,y,i}$'s, expand both sides of Eq. 19 that equation to first order in y (i.e., use implicit differentiation):

$$\begin{bmatrix} \frac{\partial \hat{\sigma}^{\zeta,y}}{\partial y} \\ \frac{\partial \mathbf{N}}{\partial y} \end{bmatrix} = - \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial \hat{\sigma}^{\zeta,y}} \frac{\partial \mathbf{f}}{\partial \mathbf{N}} \end{bmatrix}^{-1} \frac{\partial \mathbf{f}}{\partial y} \tag{21}$$

(Note that in general y is a vector, so for example $\frac{\partial \mathbf{f}}{\partial y}$, $\frac{\partial \hat{\sigma}^{\zeta,y}}{\partial y}$, and $\frac{\partial \mathbf{N}}{\partial y}$ are all matrices.) The solution to this equation gives us the partial derivatives we need to evaluate Eq. 20.

Given these partial derivatives, we employ conjugate gradient descent to update y. An alternative is to use Newton's method; to do that one needs to compute the Hessian of the QRE probabilities with respect to y, which can be done by differentiating the solution for $\frac{\partial \hat{\sigma}^{\zeta,y}}{\partial y}$ given by solving Eq. 21.

Note that in practice, when running this algorithm we can ask the players (or observe their behavior) to determine their joint mixed strategy for the current y, $\hat{\sigma}^{\zeta,y}$. So we don't have to solve the fixed point equation giving the QRE. Our descent algorithms only require that can predict the dependence on the position of the QRE on y. That means we only need to know how π and the player utilities depend on y.

E. Experimental details

DHW: Note that in our experiments, we actually have a partial information scenario, where w^i_t is player i's history of moves and rewards. However we can't write down Ω tractably, and therefore instead approximate it and its ramifications, with an exact information equilibrium model. It is that model that we then manage.

Also, somewhere we must say how in our experiments, we assume that whatever R is for the QRE that (approximates) our adaptive controllers, it is a smooth function of y.

To illustrate the various concepts outlined, we consider a simple problem of controlling a satellite in $\mathbb{R}^2$. The satellite has two controllers (players) that each fire a thruster from a set of four thrusters assigned to each of them. The resultant displacement of the satellite is given by the vector addition of the two thrusts from the two thrusters fired by the controllers. In these experiments, the dynamics of the satellite corresponds to $\pi^y(\theta_{t+1}, \theta_t, x_{t+1})$, where θ_t represents the current state of the satellite in the $\mathbb{R}^2$, and x_{t+1} is the two thrust vectors that are chosen by the two controllers to fire at the next time step. The eight possible thrust vectors are assigned an initial set of angles. Figure 1 illustrates the satellite with the thrusters. Each controller has its individual goal point where it wishes to move the satellite to. The manager has its own objective for moving the satellite.

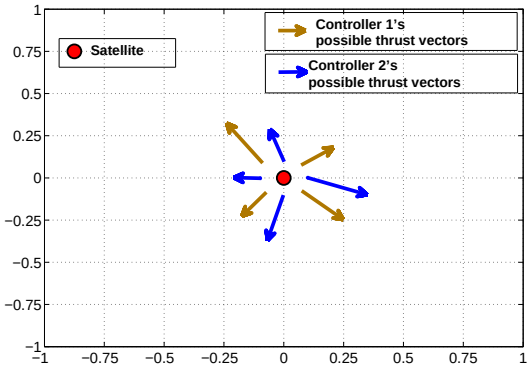

Fig. 1. Satellite Control Example

Given that the two controllers are allowed to fire one thruster each simultaneously, we implement the evolution of the satellite's trajectory as a repeated game. For this experiment, we realized the two controllers as Boltzmann reinforcement learners. Each of the four moves of both the controllers is attached a utility value. As the system trajectory evolves, the controllers update the utilities associated with each move based on how close the moves get them to their individual goals. At every time step, the controllers associate Boltzmann probabilities to the moves based on the utilities associated with those moves. Thus, moves with higher utilities are given a higher probability, and moves with lower utilities are given a lower probability. To guarantee exploration of all moves including those with low utilities, the probabilities of the individual moves have a set lower limit. If the equilibrium probabilities specified by the Boltzmann distribution fall below this limit, they are reset to this minimum threshold, and the probabilities of the remaining moves renormalized to sum to unity. This guarantees exploration of the move space along with exploitation. The reinforcement learners also use data-aging techniques to give more weight to the recent data versus old data. In these experiments, we used exponential weighting to update the utility values associated with each move. Thus, the utility value for a particular move is given by a weighted sum of the past utilities that the controller observes for that move. The exponents of the weighting term are a function of state and time. So an observed utility value that is based on the system state that is closer to the current state, and not very far back in time is given more weighting than a utility for the move that was taken at a state further away from the current state and farther back in time.

The manager, observing the behavior of the two controllers and the evolution of the system states, needs to deduce the model of the controllers' interaction. For the current experiment, the manager allows the controllers to operate without updating the thruster angles. The manager then captures the middle section of the trajectory along with the associated moves of the two controllers. The manager then estimates the rationality (β_i) of the two controllers by optimizing the log-likelihood objective function that maximizes the equilibrium probabilities of a QRE model with the observed moves. The manager, now, having a model of the interaction between the two controllers, updates the thruster angles. This update is carried out every five simulation steps. So the learning controllers operate for five simulation steps, at which point the manager updates the thruster angles. In these experiments we employed the Newton's method to update the thruster angles. Every update corresponds to multiple internal update steps, where the manager keeps on updating the angles till it can no longer increase its objective function beyond a certain prechosen threshold value.

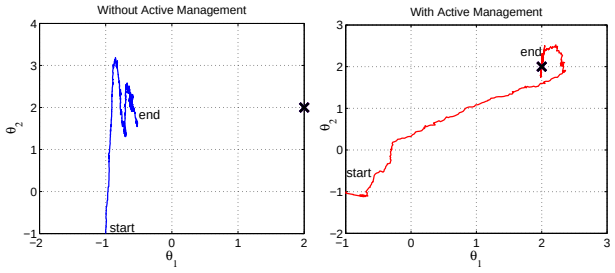

Fig. 2. Comparison of the Trajectories without (Blue) and with (Red) Active Management of the Interaction between the Learning Controllers for Case 1

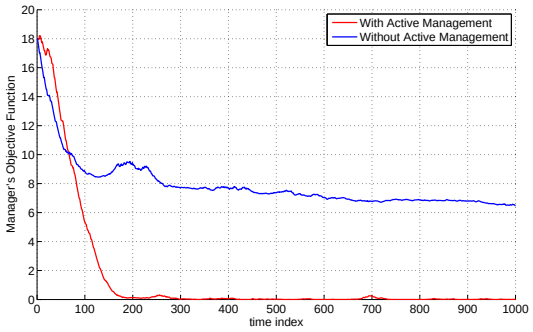

Fig. 3. Comparison of the Manager's Objective Function without (Blue) and with (Red) Active Management of the Interactions between the Learning Controllers for Case 1

In this work, we consider two cases. The first case illustrates the manager aiding the two players in achieving their common goal. In this case, the initial thruster angles for both sets of four thrusters are given to be $(85°, -85°, 95°, -95°)$ With these angles, the thrusters have limited capability in moving the satellite along the θ_1 axis. Here we set up the situation where both the controllers wish to move the satellite to the position (2,2). Figures 2 and 3 illustrate the effect of the manager, where the manager's goal is also set up to be at (2,2). Without the manager, the trajectory takes a very long time to move along the θ_1 axis. With the manager active, the thruster angles are updated to move the satellite quickly to the desired position.

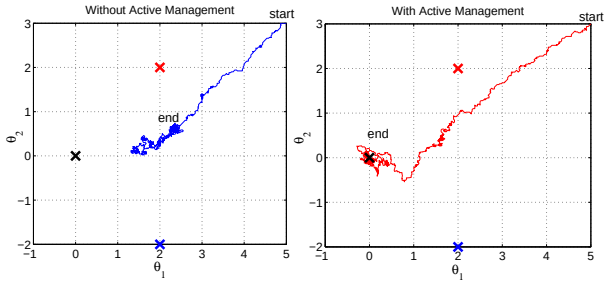

Fig. 4. Comparison of the Trajectories without (Blue) and with (Red) Active Management of the Interaction between the Learning Controllers for Case 2

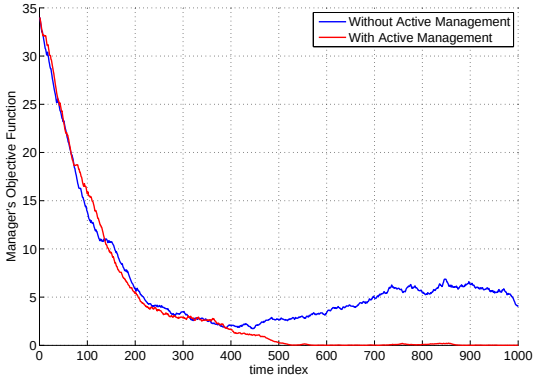

Fig. 5. Comparison of the Manager's Objective Function without (Blue) and with (Red) Active Management of the Interactions between the Learning Controllers for Case 2

In the second case, the thruster angles are given to be $(0°, 90°, 180°, 270°)$ Thus, both controllers have full controllability to move the satellite in the R^2 space. However, now controller 1's desired position is (2,2), controller 2's desired position is (2,-2), while the manager has a different goal (0,0) from these individual goals of the two controllers. Figures 4 and 5 again illustrate the trajectory of the satellite with and without active management along with the corresponding values of the manager's objective function for this case. We note that in both the cases, the manager is successful in achieving its objective.

R

[1] T. Back, D. B. Fogel, and Z. Michalewicz (eds.), *Handbook of evolutionary computation*, Oxford University Press, 1997.

[2] S. Bieniawski, I. Kroo, and D. H. Wolpert, *Flight Control with Distributed Effectors*, Proceedings of 2005 AIAA Guidance, Navigation, and Control Conference, San Francisco, CA, 2005, AIAA Paper 2005-6074.

[3] K. Chellapilla and D.B. Fogel, *Evolution, neural networks, games, and intelligence*, Proceedings of the IEEE (1999), 1471–1496.

[4] S. Choi and J.J. Alonso, *Multi-fidelity design optimization of low-boom supersonic business jet*, Proceedings of 10th AIAA/ISSMO Multidisciplany Analysis and Optimization Conference, 2004, AIAA Paper 2004-4371.

[5] E.J. Cramer, J.E. Dennis, and et alia, *Problem formulation for multidisciplinary optimization*, SIAM J. of Optimization **4** (1994).

[6] J. Ferber, *Reactive distributed artificial intelligence: Principles and applications*, Foundations of Distributed Artificial Intelligence (G. O-Hare and N. Jennings, eds.), John Wiley and Sons, 1996, pp. 287–314.

[7] D. Fudenberg and D. K. Levine, *Steady state learning and Nash equilibrium*, Econometrica **61** (1993), no. 3, 547–573.

[8] D. Fudenberg and J. Tirole, *Game theory*, MIT Press, Cambridge, MA, 1991.

[9] H. Hwang, J. Kim, and C. Tomlin, *Protocol-based conflict resolution for air traffic control*, Air Traffic Control Quarterly (2007), in press.

[10] R. D. McKelvey and T. R. Palfrey, *Quantal response equilibria for normal form games*, Games and Economic Behavior **10** (1995), 6–38.

[11] ———, *Quantal response equilibria for extensive form games*, Experimental Economics **1** (1998), 9–41.

[12] Roger B. Myerson, *Game theory: Analysis of conflict*, Harvard University Press, 1991.

[13] N. Nisan and A. Ronen, *Algorithmic mechanism design*, Games and Economic Behavior **35** (2001), 166–196.

[14] D. Rajnarayan and David H. Wolpert, *Exploiting parametric learning to improve black-box optimization*, Proceedings of ECCS 2007 (J. Jost, ed.), 2007.

[15] A. Schaerf, Y. Shoham, and M. Tennenholtz, *Adaptive load balancing: A study in multi-agent learning*, Journal of Artificial Intelligence Research **162** (1995), 475–500.

[16] J.S. Shamma and G. Arslan, *Dynamic fictitious play, dynamic gradient play, and distributed convergence to nash equilibria*, IEEE Trans. on Automatic Control **50** (2004), no. 3, 312–327.

[17] D. H. Wolpert, *Information theory - the bridge connecting bounded rational game theory and statistical physics*, Complex Engineered Systems: Science meets technology (D. Braha, A. Minai, and Y. Bar-Yam, eds.), Springer, 2004, pp. 262–290.

[18] ———, *Predicting the outcome of a game*, Submitted. See arXiv.org/abs/nlin.AO/0512015 for an early version, 2007.

[19] D. H. Wolpert and S. Bieniawski, *Distributed control by lagrangian steepest descent*, Proc. of the 2004 IEEE Control and Decision Conf., 2004, pp. 1562–1567.

[20] D. H. Wolpert, C. E. M. Strauss, and D. Rajnayaran, *Advances in distributed optimization using probability collectives*, Advances in Complex Systems (2006), in press.

[21] D. H. Wolpert and K. Tumer, *Collective Intelligence, Data Routing and Braess' Paradox*, Journal of Artificial Intelligence Research **16** (2002), 359–387.

[22] D. H. Wolpert, K. Tumer, and E. Bandari, *Improving search by using intelligent coordinates*, Physical Review E **69** (2004), no. 017701.